REMARKS

Entry of this amendment, reconsideration, and allowance are requested.

Regarding the antecedent basis objection raised with respect to claims 9, 19, and 29. The antecedent basis concerns are remedied by replacing the phrase "said address of said execute block instruction" in claims 9, 19 and 29 by the phrase "an address indicative of a memory location of said execute block instruction." Withdrawal of this objection is appropriate.

All pending claims stand rejected for obviousness based on U.S. patent 6,907,598 to Fraser in view of the common general knowledge of the skilled person. In paragraph 27 of the Office Action in the section entitled "Response to Arguments," the Examiner cites a section (pages A42-A44) of the textbook Hennessy et al., *Computer Architecture: A Quantitative Approach*, in support of the knowledge of the skilled person with respect to known techniques for exception handling. This rejection is respectfully traversed.

As argued in the earlier-filed after final response, Fraser lacks two elements recited in the independent claims. The first is "wherein when executing said block of two or more program instructions, said program counter register is configured to store an address indicative of a memory location of said execute block instruction." In the continuation sheet remarks of the advisory action, the Examiner contends that Applicant read this subject matter "as indicating that the program counter register stores the address in memory at which the execute block instruction itself is stored." Applicant does not agree. From the claim language itself, it is clear that the program counter register could store either the address of the execute block instruction, or for example, a pointer to that memory address.

The Examiner relies on col. 12, lines 14-37 with lines 20-25 of column 12 recited here for convenience:

> the PC control module 342 saves the current program counter value to a predetermined location, such as in a temporary program counter stack (not shown), and then changes the program counter based on the displacement value or parameter indicated by the Echo instruction. Modifying the program counter in this manner initiates the fetching and execution of the reference instruction or sequence of instructions (i.e. phrase), which appears somewhere in the program store before the Echo instruction (e.g. at a smaller address).

The displacement field of the Echo instruction specifies the location of the block of instructions to be executed relative to the location of the Echo instruction itself (corresponding to instruction 18 in Fraser's Figure 4). The displacement and length parameters of the Echo instructions are explicitly defined in column 9, lines 8-10; the "displacement relates to the distance back to the phrase [block of instructions]."

The text quoted from column 12 teaches that during execution of the Echo instruction, the current program counter value, which indicates the location of the Echo instruction, is saved in a temporary program counter stack. The program counter is then changed to an address indicative of the displacement value that specifies the start of the block of instructions to be repeated. The program counter value must be restored (see column 12 lines 38-39) once the block of instructions corresponding to the Echo instruction have been executed. Column 12 *teaches away from* the claimed program counter register being configured to store an address indicative of a memory location of the execute block instruction itself when executing the two or more program instructions of the block. For example see the program counter (PC) continuing to point to the address "X" corresponding to the execute block instruction (EMB) in Figure 2 of the instant application while the block of instructions are being executed. In contrast, Fraser's program counter register changes storing an address indicative of the memory location specified by the displacement value (location field) of the Echo (execute block) instruction.

The Examiner argues in the Advisory Action that "in the system of Fraser, when executing the Echo instruction, the program counter register stores an instruction address within the set of instructions [Fraser; column 12, lines 14-37]." The Examiner misunderstands the meaning of "an address indicative of a memory location of said execute block instruction." An address within the block of instructions is not an address indicative of a memory location of the execute block instruction itself. This is evident in the context of Figure 4 of Fraser where the instruction 18 corresponds to the Echo instruction, and the instructions 13-16 are four program instructions of the block or program instructions whose execution is triggered by execution of the Echo instruction. In this case, the program counter register of Fraser stores an address corresponding to the memory location of program instruction 13 during execution of the block of four block instructions and not an address indicative of the memory location of Echo instruction 18 itself. Thus, Fraser fails to teach "when executing said block of two or more program instructions, said program counter register is configured to store an address indicative of a memory location of said execute block instruction."

To clarify this point for the Examiner, the independent claims have been amended pages to specify that the program counter register stores an address indicative of a memory location of said execute block instruction "rather than to store an address indicative of said memory location specified by said location field." As simply a reinforcement of what was already claimed, this amendment is not a narrowing amendment.

The second missing claim element from Fraser is the exception handling circuit. The Examiner contends that known exception handling circuits, as described by Hennessy, could be included to save the processor execution state. The Examiner further contends that because the block count value is part of the execution state, a skilled person would necessarily provide an

exception handling circuit that stored the value of this parameter. The Examiner uses an excerpt from Hennessy to support his contentions. But Hennessy teaches on page A43 (at the start of the second paragraph following the three-itemed list):

> After the exception has been handled, special instructions return the processor from the exception by reloading the PCs and restarting the instruction stream.

Accordingly, although the skilled person might have been aware based on the Hennessy text that the program counter value is saved and restored in the event of an exception, it would not have been obvious to that person to store a block count value—in addition to a program counter value—and to restart execution of the block of program instructions corresponding to the instruction within the block that was being executed when the exception occurred. Storage of the block counter value when an exception occurs during execution of the block of two or more instructions, as specified by claim 1 involves an inventive "leap" by the skilled person relative to the disclosure of Fraser in combination with the Hennessy excerpt. Fraser does not explain what to do if an exception occurs during the Echo instruction execution.

The Examiner's motivation to modify Fraser and Hennessy is at best hindsight reconstruction. The Federal Circuit has consistently held that there must be "some teaching, suggestion, or motivation to combine references." *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998). "Stated another way, the prior art as a whole must 'suggest the desirability' of the combination." *In re Fulton*, 391 F.3d 1195, 1200 (Fed. Cir. 2004). Just because something is feasible does not make it desirable. The Federal Circuit mandates that "motivation to combine requires the latter [desirable]." *Winner Int'l Royalty Corp. v. Wang*, 202 F.3d 1340, 1349 (Fed. Cir. 2000). The rationale posited by the Examiner relates to feasibility not desirability and certainly is not found in either of the cited references.

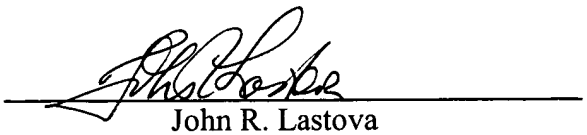
Indeed, as outlined above, since Fraser teaches that the program counter value is modified during execution of the Echo instruction, if the skilled person was to consider modifying the arrangement of Figure 3 of Fraser to include an exception handling circuit, he would consider, in the event of an exception having occurred during the Echo instruction execution and following handling of the exception, that the program counter value should be restored to the previous value by copying in the stored value from the program counter stack (see col. 12, lines 20-22). Neither Fraser nor Hennessy discloses or suggests that the value stored in the program counter register throughout execution of two or more instructions of the block is an address indicative of the memory location of the execute block instruction.

The technology defined in the independent claims provides for easier integration of execute block instructions with existing program code for exception handling. The existing program counter behavior does not need to be sufficiently altered to compensate for inclusion of the embedded block instructions. In particular, there is no need to save the current program counter to a temporary program counter stack upon execution of the execute block instruction and to restore the program counter value from a temporary program counter stack following execution of the execute block instruction. Instead, a program counter register stores an address indicative of the memory location of the execute block instruction itself so that the program counter only needs to be incremented following execution of the execute block instruction. The separate block counter register monitors which of the two or more instructions within the block is currently being executed while the value stored in the program counter register remains static during execution of the plurality of instructions of the embedded block.

The application is in condition for allowance. An early notice to that effect is respectfully submitted.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: 
John R. Lastova
Reg. No. 33,149

JRL:maa
901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100